Application No.: 10/663,632	§	Examiner: Truong, Camquy
Filed: September 16, 2003	§	Group/Art Unit: 2195
Inventor(s):	§	Atty. Dkt. No: 5760-13100/VRTS
Cuong H. Tran	§	0263
	§	
	§	
	§	
Title: Checkpoint Service	§	
	§	
	§	
	§	
	§	
	§	

REQUEST FOR PREAPPEAL BRIEF REVIEW

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request. This request is being filed with a notice of appeal. Independent claims 1, 13, and 24 are rejected under 35 U.S.C. § 102(e) as being anticipated by Lim et al., U.S. Patent No. 6,795,966 ("Lim"). Applicant sets forth the clear errors in the rejections below. Please note that for brevity, only the primary arguments directed mainly to the independent claims are presented, and that additional arguments, e.g., directed to the subject matter of the dependent claims, will be presented if and when the case proceeds to Appeal.

Applicant respectfully submits that claims 1-29 recite combinations of features not taught or suggested in the cited art. For example, claim 1 recites a combination of features including: "replicate a checkpoint segment from a first local storage of a first node to at least one other node, wherein the checkpoint segment is stored into the first local storage by an application, and wherein the checkpoint segment comprises a state of

the application."

The Office Action asserts that the checkpoint segment is anticipated by the state of the virtual machines that are checkpointed in Lim. However, **the checkpoints in Lim are made by the virtual machine monitor, and none of the virtual machine monitor state is included in the virtual machine state**. Instead, the virtual machine monitor extracts the state from the virtual machine after interrupting execution of the virtual machine: "At the completion of execution of each processor instruction by the virtual processor, the virtual machine has a total state that includes state information of the virtual processor, of the virtual memory, of the virtual operating system, of each application program, and of each virtual peripheral device. The virtual machine monitor is then used as a checkpointing mechanism a) for interrupting the virtual processor; b) for sensing a checkpoint request for the virtual machine; and c) for generating at least one checkpoint, each checkpoint comprising a list of the total state of the virtual machine corresponding to the respective checkpoint request, at the completion of execution of a respective virtual processor instruction." (Lim, col. 6, lines 40-52). Accordingly, the virtual machine monitor, outside of the virtual machine, interrupts the virtual machine and creates the checkpoint of the virtual machine state, which does not include the state of the virtual machine monitor.

The Office Action responds to the above argument in the Response to Arguments section, stating that the virtual machine is running directly on the virtual machine monitor and concluding that the checkpoints highlighted above include the state of the virtual machine monitor. See Office Action, page 9, last paragraph extending to page 10. Applicant respectfully submits that **there is no teaching in Lim that the state of the virtual machine monitor is included in the state of the virtual machine**. Instead, as noted above, Lim teaches that the state of the virtual machine includes state information of the virtual processor, the virtual memory, virtual operating system, each application program, and each virtual peripheral device. For example, Fig. 2 of Lim illustrates a VM1 (reference number 200), that includes applications 220_1-220_2, virtual operating system 202, virtual peripheral 208, virtual memory 206, and virtual processor 204. See,

e.g., Lim, col. 14, lines 27-43. **Separate from VM1 is the virtual machine monitor 250**. There is no teaching or suggestion that any state from the VMM 250 is included in the checkpoint of the VM1 or the checkpoints for any other virtual machines in Lim's system.

Additionally, claim 1 recites "replicate a checkpoint segment from a first local storage of a first node to at least one other node …and load a copy of the replicated checkpoint segment from the at least one other node into a second local storage of a second node responsive to a request from the second node to load the copy, wherein the second node is to execute the application." The Office Action relies on the language of Lim's claim 10, in col. 32, lines 44-45, to indicate that the state of the source virtual machine is replicated into secondary virtual machines. Applicant does not disagree that Lim teaches restoring the state of a source virtual machine into another virtual machine. However, Lim does not teach that the checkpoint arrives at the secondary virtual machine in the fashion recited in claim 1: from the first node's local storage to another node, and from that other node to a second node (more particularly, the second node's local storage).

The Response to Arguments section of the Office Action responds to the above argument, stating that Lim teaches transmitting a checkpoint segment from a first node to a second virtual machine monitor which restores the checkpoint into a second virtual machine, citing col. 21, lines 54-61. See Office Action, page 10, first full paragraph. Applicants respectfully submit that transmitting a checkpoint from the virtual machine monitor on the first node to the virtual machine monitor on a second node and restoring the checkpoint on the second node fails to anticipate claim 1. Only two nodes are involved in Lim's transfer: the first node and the second node. On the other hand, claim 1 transfers the checkpoint segment from the first node to at least one other node, and from that other node to a second node. Three nodes are recited in claim 1.

For at least the above stated reasons, Applicant submits that Lim does not anticipate claim 1. Claims 2-12 and 26-27 depend from claim 1 and recite additional

combinations of features not taught or suggested in Lim. Given the patentability of claim 1 over Lim, as illustrated above, further comment on claims 2-12 with regard to the rejections of those claims is unnecessary at this time. Nevertheless, Applicant notes that several of the dependent claims are rejected based on portions of Lim that are not part of Lim's system (e.g. especially references to Lim's background). For a reference to anticipate a combination of features, the reference must teach the features <u>arranged as required by the claim</u> (see MPEP 2131, last paragraph, citing *In re Bond*, 15 USPQ2d 1566 (Fed. Cir. 1990). Thus, teachings from Lim's background, which are not part of the system described in Lim's detailed description, fail to anticipate the combination of features recited in a given claim because the rejection of the independent claims relies on the teachings of Lim's system. This issue with the rejection occurs at least for claims 2-5 and 12, as well as similar claims dependent from claim 13.

Claim 13 recites a combination of features including: "an <u>application storing a checkpoint segment</u> of the application's state to a local storage of a first node on which the application is executing; <u>replicating a checkpoint segment from the first local storage of the first node to at least one other node</u>; and <u>loading a copy of the replicated checkpoint segment from the at least one other node into a second local storage of a second node</u> responsive to a request from the second node to load the copy, wherein the second node is to execute the application." The same features of Lim highlighted above with regard to claim 1 are alleged to anticipate claim 13. Applicant respectfully submits that Lim does not anticipate the above highlighted features of claim 13, either. Accordingly, claim 13 is patentable over Lim. Claims 14-23 and 28-29 depend from claim 13 and recite additional combinations of features not taught or suggested in Lim.

Claim 24 recites a combination of features including: "a first node of the plurality of nodes is configured to <u>replicate a checkpoint segment from a first local storage of the first node to at least one other node</u> of the plurality of nodes, wherein the checkpoint segment is stored into the first local storage by <u>an application executing on the first node, and wherein the checkpoint segment comprises a state of the application</u>, and wherein a second node of the plurality of nodes is configured to <u>load a copy of the replicated</u>

checkpoint segment from the at least one other node into a second local storage of the second node responsive to a request to load the copy, wherein the second node is to execute the application." The same features of Lim highlighted above with regard to claim 1 are alleged to anticipate claim 24. Applicant respectfully submits that Lim does not anticipate the above highlighted features of claim 24, either. Accordingly, claim 24 is patentable over Lim. Claims 14-23 and 28-29 depend from claim 13 and recite additional combinations of features not taught or suggested in Lim.

CONCLUSION

Applicants submit the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5760-13100/LJM.

Respectfully submitted,

/Lawrence J. Merkel/
Lawrence J. Merkel, Reg. No. 41,191
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: ___July 10, 2009___

5